UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/971,755	10/05/2001	Inderpal Singh Narang	JP920010146US1	4239

39903 7590 05/28/2004

ANTHONY V.S. ENGLAND
1717 WEST SIXTH STREET
SUITE 230
AUSTIN, TX 78703

EXAMINER	
FLEURANTIN, JEAN B	
ART UNIT	**PAPER NUMBER**
2172	

DATE MAILED: 05/28/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/971,755	NARANG ET AL.	
	Examiner	Art Unit	
	Jean B Fleurantin	2172	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *22 March 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-11,14-30,33-48 and 51-58* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-11,14-30, 33-48 and 51-58* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Response to Amendment

1. Claims 1-11, 14-30, 33-48 and 51-58 remain pending for examination.

2. The Change of Address filed on 26 March 2004 (Paper No. 11) has been entered.

Response to Arguments

3. Applicant's arguments filed 3 March 2004 have been fully considered but they are not persuasive. Because of the following reasons:

Applicant(s) stated on page 16, that amends the claims to distinctly point out and particular claim that the "object can be edited independently of the related metadata in a loose transaction model." However, Burns discloses the file management system includes a check in function which receives the updated file, and saves the updated file saves the updated file under a new name different from the original, generates new metadata for the updated file which transactionally updated file with its metadata, (see col. 4, lines 53-59), and (col. 3, lines 43-53).

In response to applicant's argument on page 19, that "Guturu '075 and Burns '064 are solving different problems than Applicant," the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988)and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992). In this case, Burns does not explicitly disclose steps of

comparing said embedded version number with a version number of a latest committed version of said externally stored object to determine if said handle refers to a current version of said externally stored object". Guturu discloses a step of comparing a version number of the data record to a version number of the data update, (see col. 2, lines 39-41), and column 4, lines 5-11-37; and also column 6, lines 47-51. It would have been obvious to one ordinary skill in the art to modify the combined teachings of Burns with Guturu with steps of comparing said embedded version number with a version number of a latest committed version of said externally stored object to determine if said handle refers to a current version of said externally stored object. Such modification would allow the teachings of Burns and Guturu to improve the accuracy and the reliability of the method of maintaining data consistency in a loose transaction model, and to provide database synchronicity without the use of synchronizing messages and thus avoiding high on the links between the databases, and to process the data update requests and maintain the same data content at the two databases, (see Guturu, col. 3, lines 4-6; and col. 3, lines 9-10).

MPEP 2111 Claim Interpretation; Broadest Reasonable Interpretation

During patent examination, the pending claims must be "given the broadest reasonable interpretation consistent with the specification" Applicant always has the opportunity to amend the claims during prosecussion and broad interpretation by the examiner reduces the possibility that the claim, once issued, will be interpreted more broadly than is justified. In re Prater, 162 USPQ 541,550-51 (CCPA 1969). The court found that applicant was advocating ... the impermissible importation of subject matter from the specification into the claim. See also In re Morris, 127 F.3d 1048, 1054-55, 44 USPQ2d 1023, 1027-28 (Fed. Cir. 1997) (The court held that the PTO is not required, in the course of prosecution, to interpret claims in applications in

the same manner as a court would interpret claims in an infringement suit. Rather, the "PTO applies to verbiage of the proposed claims the broadest reasonable meaning of the words in their ordinary usage as they would be understood by one of ordinary skill in the art, taking into account whatever enlightenment by way of definition or otherwise that may be afforded by the written description contained in application's specification.").

The broadest reasonable interpretation of the claims must also be consistent with the interpretation that those skilled in the art would reach. In re Cortright, 165 F.3d 1353, 1359, 49 USPQ2d 1464, 1468 (Fed. Cir. 1999).

For the above reasons, it is believed that the last Office Action was proper.

Claim Rejections - 35 USC § 112

4. Claims 1, 20, 39, 57 and 58 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention. The specification, as originally filed, fails to provide support for "said object capable of being edited independently of said related metadata." Page 9, lines 14-16, of the specification discloses "a loose transaction model for file and meta-data updates is useful where the file can be edited independently of the meta-data". This is not equivalent to "wherein a loose transaction model **is one** where the file can be edited independently of the metadata."

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1-11, 14-30, 33-48 and 51-58 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,088,694 issued to Burns et al. ("hereinafter Burns") in view of U.S. Patent No. 6,581,075 issued to Guturu et al. ("hereinafter Geturu").

As per claim 1, Burns discloses "a method of maintaining consistency of content of an object and metadata ~~related to said object in a loose transaction model for object and meta-data updates~~" as to maintain consistency of the file content with its metadata in the DBMS requires that certain procedure must be followed in order to modify the file, (see col. 3, lines 48-51), "storing said related ~~meta-data~~ metadata and a reference to said object in a table of a database" as the file management system also includes a function that receives the updated file, in which saves the updated file under a new name different from the original, (see col. 4, lines 54-56), and column 1, lines 55-58, "said object being stored externally to said database in an object store" as to keep large data objects stored as files in a file system and link these references to these external files from the database (col. 2, lines 14-16), "said object capable of being edited independently of said related metadata" as the file management system includes a check in function which receives the updated file, and saves the updated file saves the updated file under a new name different from the original, generates new metadata for the updated file which

transactionally updated file with its metadata, (see col. 4, lines 53-59), and (col. 3, lines 43-53), "said reference used to obtain a handle for directly accessing or manipulating said external object" as a file manager when executed by the computer system that accesses files maintained in file storage and communicates with a database management system that supports the linking of external files, (see col. 20, lines 28-32) and column 7, lines 15-17;

"obtaining a version number embedded in said handle in an access token" as the reference file A(1) is referred to as a versioned file, in which file A(1) is backed up in total, the backup operation of the new version file A(2) would involve backing up only the modified portions with respect to A(1), (see col. 5, lines 63-67). Burns does not explicitly disclose steps of comparing said embedded version number with a version number of a latest committed version of said externally stored object to determine if said handle refers to a current version of said externally stored object". Guturu discloses a step of comparing a version number of the data record to a version number of the data update, (see col. 2, lines 39-41), and column 4, lines 5-11-37; and also column 6, lines 47-51. It would have been obvious to one ordinary skill in the art to modify the combined teachings of Burns with Guturu with steps of comparing said embedded version number with a version number of a latest committed version of said externally stored object to determine if said handle refers to a current version of said externally stored object. Such modification would allow the teachings of Burns and Guturu to improve the accuracy and the reliability of the method of maintaining data consistency in a loose transaction model, and to provide database synchronicity without the use of synchronizing messages and thus avoiding high on the links between the databases, and to process the data update requests and maintain the same data content at the two databases, (see Guturu, col. 3, lines 4-6; and col. 3, lines 9-10).

As per claim 2, in addition to the discussion in claim 1, Burns does not explicitly discloses steps of comparing a last modification time stamp of said object with a last modification timestamp for said latest committed version of said object; and if said last modification time stamp of said object matches with said last modification timestamp for said latest committed version of said object, permitting access to said externally stored object. However, Guturu discloses "comparing a last modification time stamp of said object with a last modification timestamp for said latest committed version of said object" as means for receiving a data update request for a data record at the database, and comparing a timestamp of the data record to a timestamp of the data update request, (see col. 2, lines 7-10); and "if said last modification time stamp of said object matches with said last modification timestamp for said latest committed version of said object" as if the timestamp of the data update request is substantially identical to the timestamp of the data record, in which the operational priority of the data record is compared with the operational priority of the data update request, (see col. 2, lines 48-52), "permitting access to said externally stored object" as the delete flag is set to false the override SMS flag is set to false, in which the insert flag is set to true, and the timestamp is set to the timestamp of the network side update request (see col. 4, lines 64-67), and column 2, lines 63-65. It would have been obvious to one ordinary skill in the art to modify the combined teachings of Burns with Guturu with steps of comparing a last modification time stamp of said object with a last modification timestamp for said latest committed version of said object; and if said last modification time stamp of said object matches with said last modification timestamp for said latest committed version of said object, permitting access to said externally stored object.

Such modification would allow the teachings of Burns and Guturu to improve the accuracy and the reliability of the method of maintaining data consistency in a loose transaction model, and to provide a method of maintaining synchronization among multiple databases, (see Guturu, col. 1, lines 57-58).

As per claim 3, Burns discloses the claimed subject matter except the claimed if said last modification time stamp of said object does not match with said last modification timestamp for said latest committed version of said object, generating an error to indicate that said handle refers to stale content in said object. However, Guturu discloses the update record timestamp is compared to the existing record timestamp, if the difference is greater than the predetermined conflict time window, then a one or true is returned to the determination made, if the existing record timestamp is a conflict time window more than the updated record timestamp, then a zero or false returned, (see col. 6, lines 3-16). It would have been obvious to one ordinary skill in the art to modify the combined teachings of Burns with Guturu with steps if said last modification time stamp of said object does not match with said last modification timestamp for said latest committed version of said object, generating an error to indicate that said handle refers to stale content in said object. Such modification would allow the teachings of Burns and Guturu to improve the accuracy and the reliability of the method of maintaining data consistency in a loose transaction model, and to provide database synchronicity without the use of synchronizing messages and thus avoiding high on the links between the databases, and to process the data update requests and maintain the same data content at the two databases, (see Guturu, col. 3, lines 4-6; and col. 3, lines 9-10).

As per claim 4, Burns discloses, "the method further including the steps of updating said object in-place under either DBMS control or file system control and linking said ~~meta-data~~ metadata and said object under DBMS control" as linked a database management system through a datalink data type, even while the file is being modified at the file management system with either the update operation, (see col. 4, lines 32-36).

As per claim 5, Burns discloses, "wherein ~~said loose transaction update model~~ said object capable of being edited independently of said related metadata" as the file management system includes a check in function" as the file management system includes a check in function which receives the updated file saves the updated file saves the updated file under a new name different from the original, generates new metadata for the updated file which transactionally updated file with its metadata, (see col. 4, lines 53-59), and (col. 3, lines 43-53)," and uses SQL Mediated Object Manipulation (SMOM) for an object that resides external to said database" as the computing system issues an SQL insert, SQL delete or SQL update call in the database, in which the database management detects that this operation occurs on a column of type datalink and issues a linkfile command, (see col. 9, lines 1-6).

As per claim 6, Burns discloses the claimed subject matter except the claimed intercepting a native access to said externally stored object or a file system and validating the caller's access rights based on a combination of said version number and a last modification timestamp for a version of said object. However, Guturu discloses the update record timestamp

is compared to the existing record timestamp, if the difference is greater than the predetermined conflict time window, then a one or true is returned to the determination made, if the existing record timestamp is a conflict time window more than the updated record timestamp, then a zero or false is returned, (see col. 6, lines 3-16), and column 4, lines 64-67. It would have been obvious to one ordinary skill in the art to modify the combined teachings of Burns with Guturu with steps of intercepting a native access to said externally stored object or a file system and validating the caller's access rights based on a combination of said version number and a last modification timestamp for a version of said object. Such modification would allow the teachings of Burns and Guturu to improve the accuracy and the reliability of the method of maintaining data consistency in a loose transaction model, and to provide database synchronicity without the use of synchronizing messages and thus avoiding high on the links between the databases, and to process the data update requests and maintain the same data content at the two databases, (see Guturu, col. 3, lines 4-6; and col. 3, lines 9-10).

As per claim 7, Burns discloses, "wherein said intercepting step is carried out using a filter layer of said object store for said stored object", (see figure 6, element 102, col. 10, lines 35-37).

As per claim 8, Burns discloses, "wherein said object store is a local file system", (see col. 4, lines 54-56).

As per claim 9, Burns discloses, "wherein said object store is a distributed file system, said object being accessed from a remote file system client" as the file with the name filename is created in the file server 17 and accessed by the client application 80 over the file communication path, (see col. 9, lines 42-44).

As per claim 10, Burns discloses, "wherein a file access occurs in the presence of authoritative caching and said comparing steps are performed at ~~saidfile~~ said file system client" as the file with the name filename is created in the file server 17 and accessed by the client application 80 over the file communication path, in which the client application sends a request over the SQL communication path for the insertion of a record with a datalink field containing a server/filename entry into the database stored at the database storage, (see col. 9, lines 42-48).

As per claim 11, Burns discloses the claimed subject matter except the claimed caching the last known version number and the corresponding last modification timestamp at said file system client after an access and refreshing said last known version number and said corresponding last modification timestamp with latest values from a file server the next time one or both of said comparisons fail with the previously cached values, in which case said comparing steps are retried with refreshed values. However, Guturu discloses "caching the last known version number and the corresponding last modification timestamp at said file system client after an access and refreshing said last known version number" as the method includes the steps of receiving a data update request for a data record at the database, and comparing a timestamp of the data record to a timestamp of the data update request, (see col. 2, lines 7-10), "and said

corresponding last modification timestamp with latest values from a file server the next time one or both of said comparisons fail with the previously cached values, in which case said comparing steps are retried with refreshed values" as the method then updates the data in the data record with the data update request if the timestamp of the data update request is a predetermined conflict time window later than the timestamp of the data record, in which the operational priority of the data record is then compared to the operational priority of the data update request if the timestamp of the data update request is substantially identical to the timestamp of the data record, and the data in the data record is updated with the data update request if the operational priority of the data update request is higher than the operational priority of the data record, (see col. 2, lines 10-20). Further, in column 2, lines 63-65, Guturu discloses the data update request is ignored in response to the timestamp of the data update request being substantially identical to the timestamp of the data record. It would have been obvious to one ordinary skill in the art to modify the combined teachings of Burns with Guturu with steps of caching the last known version number and the corresponding last modification timestamp at said file system client after an access and refreshing said last known version number and said corresponding last modification timestamp with latest values from a file server the next time one or both of said comparisons fail with the previously cached values, in which case said comparing steps are retried with refreshed values. Such modification would allow the combined teachings of Burns and Guturu to improve the accuracy and the reliability of the method of maintaining data consistency in a loose transaction model, and to provide database synchronicity without the use of synchronizing messages and thus avoiding high on the links between the databases, and to

process the data update requests and maintain the same data content at the two databases, (see Guturu, col. 3, lines 4-6; and col. 3, lines 9-10).

As per claim 14, Burns discloses, "wherein said version number is temporally unique" as the reference file A (1) is referred to as a versioned file, (see col. 5, lines 64-65).

As per claim 15, Burns discloses the claimed subject matter except the claimed wherein the last-modification-timestamp attribute associated with said object is maintained by said object store. However, Guturu discloses the steps of receiving a data update request for a data record at the database and comparing a timestamp of the data request, (see col. 2, lines 7-10). It would have been obvious to one ordinary skill in the art to modify the combined teachings of Burns with Guturu with steps of wherein the last-modification-timestamp attribute associated with said object is maintained by said object store. Such modification would allow the teachings of Burns and Guturu to improve the accuracy and the reliability of the method of maintaining data consistency in a loose transaction model, and to provide database synchronicity without the use of synchronizing messages and thus avoiding high on the links between the databases, and to process the data update requests and maintain the same data content at the two databases, (see Guturu, col. 3, lines 4-6; and col. 3, lines 9-10).

As per claim 16, Burns discloses, "wherein clock synchronization between a database server and a filesystem server is not required", (see col. 2, lines 29-35).

As per claim 17, Burns discloses, "wherein said database is rolled back to an earlier state", (see col. 5, lines 63-67).

As per claim 18, Burns discloses, "wherein said database is a replicated version", (see col. 5, lines 63-67).

As per claim 19, Burns discloses the method, "further including the steps of updating said object while said object is currently linked" as user updates a linked file, that provided with a copy of the original linked file to maintain as local copy for updating, which during such updating operations the original unmodified reference file version remains in the file storage, (see col. 12, lines 57-61); and

"accessing said ~~meta-data~~ metadata for said object while said object is being updated" as access file data through local file systems and make modifications to that data, (see col. 4, lines 39-40) .

As per claim 20, Burns discloses "an apparatus for maintaining consistency of content of an object and metadata related to said object ~~in a loose transaction model for object and meta-data updates~~" as to maintain consistency of the file content with its metadata in the DBMS requires that certain procedure must be followed in order to modify the file, (see col. 3, lines 48-51), "means for storing said related ~~meta-data~~ metadata and a reference to said object in a table of a database" as the file management system also includes a function that receives the updated file, in which saves the updated file under a new name different from the original, (see col. 4,

lines 54-56), and column 1, lines 55-58, "said object being stored externally to said database in an object store" as to keep large data objects stored as files in a file system and link these references to these external files from the database (col. 2, lines 14-16), and "said object capable of being edited independently of said related metadata in a loose transaction model" as the file management system includes a check in function which receives the updated file, and saves the updated file saves the updated file under a new name different from the original, generates new metadata for the updated file which transactionally updated file with its metadata, (see col. 4, lines 53-59), and (col. 3, lines 43-53), "said reference used to obtain a handle for directly accessing or manipulating said external object" as a file manager when executed by the computer system that accesses files maintained in file storage and communicates with a database management system that supports the linking of external files, (see col. 20, lines 28-32) and column 7, lines 15-17;

"means for obtaining a version number embedded in said handle" as the reference file A(1) is referred to as a versioned file, in which file A(1) is backed up in total, the backup operation of the new version file A(2) would involve backing up only the modified portions with respect to A(1), (see col. 5, lines 63-67). Burns does not explicitly disclose steps means for comparing said embedded version number with a version number of a latest committed version of said externally stored object to determine if said handle refers to a current version of said externally stored object" Guturu discloses means for comparing a version number of the data record to a version number of the data update, (see col. 2, lines 39-41), and column 4, lines 5-11-37; and also column 6, lines 47-51. It would have been obvious to one ordinary skill in the art to modify the combined teachings of Burns with Guturu with steps of comparing said embedded

version number with a version number of a latest committed version of said externally stored object to determine if said handle refers to a current version of said externally stored object. Such modification would allow the teachings of Burns and Guturu to improve the accuracy and the reliability of the method of maintaining data consistency in a loose transaction model, and to provide database synchronicity without the use of synchronizing messages and thus avoiding high on the links between the databases, and to process the data update requests and maintain the same data content at the two databases, (see Guturu, col. 3, lines 4-6; and col. 3, lines 9-10).

As per claim 21, in addition to the discussion in claim 20, Burns does not explicitly discloses steps of comparing a last modification time stamp of said object with a last modification timestamp for said latest committed version of said object; and means for, if said last modification time stamp of said object matches with said last modification timestamp for said latest committed version of said object, permitting access to said externally stored object. However, Guturu discloses "comparing a last modification time stamp of said object with a last modification timestamp for said latest committed version of said object" as means for receiving a data update request for a data record at the database, and comparing a timestamp of the data record to a timestamp of the data update request, (see col. 2, lines 7-10); and "means for, if said last modification time stamp of said object matches with said last modification timestamp for said latest committed version of said object" as if the timestamp of the data update request is substantially identical to the timestamp of the data record, in which the operational priority of the data record is compared with the operational priority of the data update request, (see col. 2, lines 48-52), "permitting access to said externally stored object" as the delete flag is set to false the

override SMS flag is set to false, in which the insert flag is set to true, and the timestamp is set to the timestamp of the network side update request (see col. 4, lines 64-67), and column 2, lines 63-65. It would have been obvious to one ordinary skill in the art to modify the combined teachings of Burns with Guturu with steps of comparing a last modification time stamp of said object with a last modification timestamp for said latest committed version of said object; and if said last modification time stamp of said object matches with said last modification timestamp for said latest committed version of said object, permitting access to said externally stored object. Such modification would allow the teachings of Burns and Guturu to improve the accuracy and the reliability of the method of maintaining data consistency in a loose transaction model, and to provide database synchronicity without the use of synchronizing messages and thus avoiding high on the links between the databases, and to provide a method of maintaining synchronization among multiple databases, (see Guturu, col. 1, lines 57-58).

As per claim 22, Burns discloses the claimed subject matter except the claimed if said last modification time stamp of said object does not match with said last modification timestamp for said latest committed version of said object, generating an error to indicate that said handle refers to stale content in said object. However, Guturu discloses the update record timestamp is compared to the existing record timestamp, if the difference is greater than the predetermined conflict time window, then a one or true is returned to the determination made, if the existing record timestamp is a conflict time window more than the updated record timestamp, then a zero or false returned, (see col. 6, lines 3-16). It would have been obvious to one ordinary skill in the art to modify the combined teachings of Burns with Guturu with steps if said last modification

time stamp of said object does not match with said last modification timestamp for said latest committed version of said object, generating an error to indicate that said handle refers to stale content in said object. Such modification would allow the teachings of Burns and Guturu to improve the accuracy and the reliability of the method of maintaining data consistency in a loose transaction model, and to provide database synchronicity without the use of synchronizing messages and thus avoiding high on the links between the databases, and to process the data update requests and maintain the same data content at the two databases, (see Guturu, col. 3, lines 4-6; and col. 3, lines 9-10).

As per claim 23, Burns discloses, "the method further including means for updating said object in-place under either DBMS control or file system control and linking said ~~meta-data~~ metadata and said object under DBMS control" as linked a database management system through a datalink data type, even while the file is being modified at the file management system with either the update operation, (see col. 4, lines 32-36).

As per claim 24, Burns discloses, "wherein said loose-transaction update model uses SQL Mediated Object Manipulation (SMOM) for an object that resides external to said database" as the computing system issues an SQL insert, SQL delete or SQL update call in the database, in which the database management detects that this operation occurs on a column of type datalink and issues a linkfile command, (see col. 9, lines 1-6).

As per claim 25, Burns discloses the claimed subject matter except the claimed intercepting a native access to said externally stored object or a file system and validating the caller's access rights based on a combination of said version number and a last modification timestamp for a version of said object. However, Guturu discloses the update record timestamp is compared to the existing record timestamp, if the difference is greater than the predetermined conflict time window, then a one or true is returned to the determination made, if the existing record timestamp is a conflict time window more than the updated record timestamp, then a zero or is false returned, (see col. 6, lines 3-16), and column 4, lines 64-67. It would have been obvious to one ordinary skill in the art to modify the combined teachings of Burns with Guturu with steps of intercepting a native access to said externally stored object or a file system and validating the caller's access rights based on a combination of said version number and a last modification timestamp for a version of said object. Such modification would allow the teachings of Burns and Guturu to improve the accuracy and the reliability of the method of maintaining data consistency in a loose transaction model, and to provide database synchronicity without the use of synchronizing messages and thus avoiding high on the links between the databases, and to process the data update requests and maintain the same data content at the two databases, (see Guturu, col. 3, lines 4-6; and col. 3, lines 9-10).

As per claim 26, Burns discloses, "wherein said intercepting step is carried out using a filter layer of said object store for said stored object", (see figure 6, element 102, col. 10, lines 35-37).

As per claim 27, Burns discloses, "wherein said object store is a local file system", (see col. 4, lines 54-56).

As per claim 28, Burns discloses, "wherein said object store is a distributed file system, said object being accessed from a remote file system client", (see col. 9, lines 42-44).

As per claim 29, Burns discloses, wherein a file access occurs in the presence of authoritative caching and both said means for comparing are implemented at said file system client, (see col. 4, lines 54-56).

As per claim 30, Burns discloses the claimed subject matter except the claimed means for caching the last known version number and the corresponding last modification timestamp at said file system client after an access and means for refreshing said last known version number and said corresponding last modification timestamp with latest values from a file server the next time one or both of said comparisons fail with the previously cached values, in which case said comparing means retry said comparisons with refreshed values. However, Guturu discloses "means for caching the last known version number and the corresponding last modification timestamp at said file system client after an access and means for refreshing said last known version number" as the method includes the steps of receiving a data update request for a data record at the database, and comparing a timestamp of the data record to a timestamp of the data update request, (see col. 2, lines 7-10), "and said corresponding last modification timestamp with latest values from a file server the next time one or both of said comparisons fail with the

previously cached values, in which case both comparing mean retry with refreshed values" as the method then updates the data in the data record with the data update request if the timestamp of the data update request is a predetermined conflict time window later than the timestamp of the data record, in which the operational priority of the data record is then compared to the operational priority of the data update request if the timestamp of the data update request is substantially identical to the timestamp of the data record, and the data in the data record is updated with the data update request if the operational priority of the data update request is higher than the operational priority of the data record, (see col. 2, lines 10-20). Further, in column 2, lines 63-65, Guturu discloses the data update request is ignored in response to the timestamp of the data update request being substantially identical to the timestamp of the data record. It would have been obvious to one ordinary skill in the art to modify the combined teachings of Burns with Guturu with steps of caching the last known version number and the corresponding last modification timestamp at said file system client after an access and refreshing said last known version number and said corresponding last modification timestamp with latest values from a file server the next time one or both of said comparisons fail with the previously cached values, in which case said comparing steps are retried with refreshed values. Such modification would allow the combined teachings of Burns and Guturu to improve the accuracy and the reliability of the method of maintaining data consistency in a loose transaction model, and to provide database synchronicity without the use of synchronizing messages and thus avoiding high on the links between the databases, and to process the data update requests and maintain the same data content at the two databases, (see Guturu, col. 3, lines 4-6; and col. 3, lines 9-10).

As per claim 33, Burns discloses, "wherein said version number is temporally unique" as the reference file A (1) is referred to as a versioned file, (see col. 5, lines 64-65).

As per claim 34, Burns discloses the claimed subject matter except the claimed wherein the last-modification-timestamp attribute associated with said object is maintained by said object store. However, Guturu discloses the steps of receiving a data update request for a data record at the database and comparing a timestamp of the data request, (see col. 2, lines 7-10). It would have been obvious to one ordinary skill in the art to modify the combined teachings of Burns with Guturu with steps of wherein the last-modification-timestamp attribute associated with said object is maintained by said object store. Such modification would allow the teachings of Burns and Guturu to improve the accuracy and the reliability of the method of maintaining data consistency in a loose transaction model, and to provide database synchronicity without the use of synchronizing messages and thus avoiding high on the links between the databases, and to process the data update requests and maintain the same data content at the two databases, (see Guturu, col. 3, lines 4-6; and col. 3, lines 9-10).

As per claim 35, Burns discloses, "wherein clock synchronization between a database server and a filesystem server is not required", (see col. 2, lines 29-35).

As per claim 36, Burns discloses, "wherein said database is rolled back to an earlier state", (see col. 5, lines 63-67).

As per claim 37, Burns discloses, "wherein said database is a replicated version", (see col. 5, lines 63-67).

As per claim 38, Burns discloses the method, "means for updating said object while said object is currently linked" as user updates a linked file, that provided with a copy of the original linked file to maintain as local copy for updating, which during such updating operations the original unmodified reference file version remains in the file storage, (see col. 12, lines 57-61); and

"means for accessing said ~~meta-data~~ metadata for said object while said object is being updated" as access file data through local file systems and make modifications to that data, (see col. 4, lines 39-40).

As per claim 39, Burns discloses "a computer program of maintaining consistency of content of an object and metadata related to said object ~~in a loose transaction model for object and meta-data updates~~" as to maintain consistency of the file content with its metadata in the DBMS requires that certain procedure must be followed in order to modify the file, (see col. 3, lines 48-51), "computer code for storing said related ~~meta-data~~ metadata and a reference to said object in a table of a database" as the file management system also includes a function that receives the updated file, in which saves the updated file under a new name different from the original, (see col. 4, lines 54-56), and column 1, lines 55-58, "said object being stored externally to said database in an object store" as to keep large data objects stored as files in a file system

and link these references to these external files from the database (col. 2, lines 14-16), and "said object capable of being edited independently of said related metadata in a loose transaction model" as the file management system includes a check in function which receives the updated file saves the updated file, and saves the updated file under a new name different from the original, generates new metadata for the updated file which transactionally updated file with its metadata, (see col. 4, lines 53-59), and (col. 3, lines 43-53), "said reference used to obtain a handle for directly accessing or manipulating said external object" as a file manager when executed by the computer system that accesses files maintained in file storage and communicates with a database management system that supports the linking of external files, (see col. 20, lines 28-32) and column 7, lines 15-17;

"computer code for obtaining a version number embedded in an access token in said handle" as the reference file A(1) is referred to as a versioned file, in which file A(1) is backed up in total, the backup operation of the new version file A(2) would involve backing up only the modified portions with respect to A(1), (see col. 5, lines 63-67). Burns does not explicitly disclose steps computer code for comparing said embedded version number with a version number of a latest committed version of said externally stored object to determine if said handle refers to a current version of said externally stored object" Guturu discloses step of comparing a version number of the data record to a version number of the data update, (see col. 2, lines 39-41), and column 4, lines 5-11-37; and also column 6, lines 47-51. It would have been obvious to one ordinary skill in the art to modify the combined teachings of Burns with Guturu with steps of comparing said embedded version number with a version number of a latest committed version of said externally stored object to determine if said handle refers to a current version of said

externally stored object. Such modification would allow the teachings of Burns and Guturu to improve the accuracy and the reliability of the method of maintaining data consistency in a loose transaction model, and to provide database synchronicity without the use of synchronizing messages and thus avoiding high on the links between the databases, and to process the data update requests and maintain the same data content at the two databases, (see Guturu, col. 3, lines 4-6; and col. 3, lines 9-10).

As per claim 40, in addition to the discussion in claim 39, Burns does not explicitly discloses steps of comparing a last modification time stamp of said object with a last modification timestamp for said latest committed version of said object; and means for, if said last modification time stamp of said object matches with said last modification timestamp for said latest committed version of said object, permitting access to said externally stored object. However, Guturu discloses "comparing a last modification time stamp of said object with a last modification timestamp for said latest committed version of said object" as means for receiving a data update request for a data record at the database, and comparing a timestamp of the data record to a timestamp of the data update request, (see col. 2, lines 7-10); and "computer code for, if said last modification time stamp of said object matches with said last modification timestamp for said latest committed version of said object" as if the timestamp of the data update request is substantially identical to the timestamp of the data record, in which the operational priority of the data record is compared with the operational priority of the data update request, (see col. 2, lines 48-52), "permitting access to said externally stored object" as the delete flag is set to false the override SMS flag is set to false, in which the insert flag is set to true, and the

timestamp is set to the timestamp of the network side update request (see col. 4, lines 64-67), and column 2, lines 63-65. It would have been obvious to one ordinary skill in the art to modify the combined teachings of Burns with Guturu with steps of comparing a last modification time stamp of said object with a last modification timestamp for said latest committed version of said object; and if said last modification time stamp of said object matches with said last modification timestamp for said latest committed version of said object, permitting access to said externally stored object. Such modification would allow the teachings of Burns and Guturu to improve the accuracy and the reliability of the method of maintaining data consistency in a loose transaction model, and to provide database synchronicity without the use of synchronizing messages and thus avoiding high on the links between the databases, and to provide a method of maintaining synchronization among multiple databases, (see Guturu, col. 1, lines 57-58).

As per claim 41, Burns discloses the claimed subject matter except the claimed if said last modification time stamp of said object does not match with said last modification timestamp for said latest committed version of said object, generating an error to indicate that said handle refers to stale content in said object. However, Guturu discloses the update record timestamp is compared to the existing record timestamp, if the difference is greater than the predetermined conflict time window, then a one or true is returned to the determination made, if the existing record timestamp is a conflict time window more than the updated record timestamp, then a zero or false returned, (see col. 6, lines 3-16). It would have been obvious to one ordinary skill in the art to modify the combined teachings of Burns with Guturu with steps if said last modification time stamp of said object does not match with said last modification timestamp for said latest

committed version of said object, generating an error to indicate that said handle refers to stale content in said object. Such modification would allow the teachings of Burns and Guturu to improve the accuracy and the reliability of the method of maintaining data consistency in a loose transaction model, and to provide database synchronicity without the use of synchronizing messages and thus avoiding high on the links between the databases, and to process the data update requests and maintain the same data content at the two databases, (see Guturu, col. 3, lines 4-6; and col. 3, lines 9-10).

As per claim 42, Burns discloses, "computer code updating said object in-place under either DBMS control or file system control and linking said ~~meta-data~~ metadata and said object under DBMS control" as linked a database management system through a datalink data type, even while the file is being modified at the file management system with either the update operation, (see col. 4, lines 32-36).

As per claim 43, Burns discloses, "wherein said loose-transaction update model uses SQL Mediated Object Manipulation (SMOM) for an object that resides external to said database" as the computing system issues an SQL insert, SQL delete or SQL update call in the database, in which the database management detects that this operation occurs on a column of type datalink and issues a linkfile command, (see col. 9, lines 1-6).

As per claim 44, Burns discloses the claimed subject matter except the claimed computer code for intercepting a native access to said externally stored object or a file system and

validating the caller's access rights based on a combination of said version number and a last modification timestamp for a version of said object. However, Guturu discloses the update record timestamp is compared to the existing record timestamp, if the difference is greater than the predetermined conflict time window, then a one or true is returned to the determination made, if the existing record timestamp is a conflict time window more than the updated record timestamp, then a zero or false is returned, (see col. 6, lines 3-16), and column 4, lines 64-67. It would have been obvious to one ordinary skill in the art to modify the combined teachings of Burns with Guturu with steps of intercepting a native access to said externally stored object or a file system and validating the caller's access rights based on a combination of said version number and a last modification timestamp for a version of said object. Such modification would allow the teachings of Burns and Guturu to improve the accuracy and the reliability of the method of maintaining data consistency in a loose transaction model, and to provide database synchronicity without the use of synchronizing messages and thus avoiding high on the links between the databases, and to process the data update requests and maintain the same data content at the two databases, (see Guturu, col. 3, lines 4-6; and col. 3, lines 9-10).

As per claim 45, Burns discloses, "wherein said intercepting step is carried out using a filter layer of said object store for said stored object", (see figure 6, element 102, col. 10, lines 35-37).

As per claim 46, Burns discloses, "wherein said object store is a local file system", (see col. 4, lines 54-56).

As per claim 47, Burns discloses, "wherein said object store is a distributed file system, said object being accessed from a remote file system client" as the file with the name filename is created in the file server 17 and accessed by the client application 80 over the file communication path, (see col. 9, lines 42-44).

As per claim 48, Burns discloses, "wherein a file access occurs in the presence of authoritative caching and said comparing steps are performed at said file system client" as the file with the name filename is created in the file server 17 and accessed by the client application 80 over the file communication path, in which the client application sends a request over the SQL communication path for the insertion of a record with a datalink field containing a server/filename entry into the database stored at the database storage, (see col. 9, lines 42-48).

As per claim 51, Burns discloses, "wherein said version number is temporally unique" as the reference file A (1) is referred to as a versioned file, (see col. 5, lines 64-65).

As per claim 52, Burns discloses the claimed subject matter except the claimed wherein the last-modification-timestamp attribute associated with said object is maintained by said object store. However, Guturu discloses the steps of receiving a data update request for a data record at the database and comparing a timestamp of the data request, (see col. 2, lines 7-10). It would have been obvious to one ordinary skill in the art to modify the combined teachings of Burns with Guturu with steps of wherein the last-modification-timestamp attribute associated with said

object is maintained by said object store. Such modification would allow the teachings of Burns and Guturu to improve the accuracy and the reliability of the method of maintaining data consistency in a loose transaction model, and to provide database synchronicity without the use of synchronizing messages and thus avoiding high on the links between the databases, and to process the data update requests and maintain the same data content at the two databases, (see Guturu, col. 3, lines 4-6; and col. 3, lines 9-10).

As per claim 53, Burns discloses, "wherein clock synchronization between a database server and a filesystem server is not required", (see col. 2, lines 29-35).

As per claim 54, Burns discloses, "wherein said database is rolled back to an earlier state", (see col. 5, lines 63-67).

As per claim 55, Burns discloses, "wherein said database is a replicated version", (see col. 5, lines 63-67).

As per claim 56, Burns discloses the method, "means for updating said object while said object is currently linked" as user updates a linked file, that provided with a copy of the original linked file to maintain as local copy for updating, which during such updating operations the original unmodified reference file version remains in the file storage, (see col. 12, lines 57-61); and

"means for accessing said ~~meta-data~~ metadata for said object while said object is being updated" as access file data through local file systems and make modifications to that data, (see col. 4, lines 39-40).

As per claim 57, Burns discloses "a computer program product having a computer readable medium having a computer program recorded therein for maintaining consistency of content of an object and metadata related to said object ~~in a loose transaction model for object and meta-data updates~~" as to maintain consistency of the file content with its metadata in the DBMS requires that certain procedure must be followed in order to modify the file, (see col. 3, lines 48-51), "said computer program product including computer program code means for storing said related ~~meta-data~~ metadata and a reference to said object in a table of a database" as the file management system also includes a function that receives the updated file, in which saves the updated file under a new name different from the original, (see col. 4, lines 54-56), and column 1, lines 55-58, "said object being stored externally to said database in an object store" as to keep large data objects stored as files in a file system and link these references to these external files from the database (col. 2, lines 14-16), and "said object capable of being edited independently of said related metadata in a loose transaction model" as the file management system includes a check in function which receives the updated file, and saves the updated file saves the updated file under a new name different from the original, generates new metadata for the updated file which transactionally updated file with its metadata, (see col. 4, lines 53-59), and (col. 3, lines 43-53), "said reference used to obtain a ~~handle~~ filename of said object, said filename having an encrypted access token having a hash value containing a version number"

(see col. 14, lines 16-22) "for directly accessing or manipulating said external object" as a file manager when executed by the computer system that accesses files maintained in file storage and communicates with a database management system that supports the linking of external files, (see col. 20, lines 28-32) and column 7, lines 15-17;

"computer program code means for obtaining a version number embedded in said ~~handle~~ hash value" as the reference file A(1) is referred to as a versioned file, in which file A(1) is backed up in total, the backup operation of the new version file A(2) would involve backing up only the modified portions with respect to A(1), (see col. 5, lines 63-67). Burns does not explicitly disclose steps computer program code means for comparing said embedded version number with a version number of a latest committed version of said externally stored object to determine if said ~~handle~~ embedded version number refers to a current version of said externally stored object. Guturu discloses step of comparing a version number of the data record to a version number of the data update, (see col. 2, lines 39-41), and column 4, lines 5-11-37; and also column 6, lines 47-51. It would have been obvious to one ordinary skill in the art to modify the combined teachings of Burns with Guturu with steps of comparing said embedded version number with a version number of a latest committed version of said externally stored object to determine if said ~~handle~~ embedded version number refers to a current version of said externally stored object. Such modification would allow the teachings of Burns and Guturu to improve the accuracy and the reliability of the method of maintaining data consistency in a loose transaction model, and to provide database synchronicity without the use of synchronizing messages and thus avoiding high on the links between the databases, and to process the data update requests

and maintain the same data content at the two databases, (see Guturu, col. 3, lines 4-6; and col. 3, lines 9-10).

As per claim 58, Burns discloses "a system for maintaining consistency of content of an object and metadata related to said object ~~in a loose transaction model for object and meta-data updates~~" as to maintain consistency of the file content with its metadata in the DBMS requires that certain procedure must be followed in order to modify the file, (see col. 3, lines 48-51), "a database storing said related ~~meta-data~~ metadata and a reference to said object in a table of a database" as the file management system also includes a function that receives the updated file, in which saves the updated file under a new name different from the original, (see col. 4, lines 54-56), and column 1, lines 55-58, "said object being stored externally to said database in an object store" as to keep large data objects stored as files in a file system and link these references to these external files from the database (col. 2, lines 14-16), "said reference used to obtain a ~~handle~~ an encrypted access token having a hash value with an embedded version number" (see col. 14, lines 16-22) "for directly accessing or manipulating said external object" as a file manager when executed by the computer system that accesses files maintained in file storage and communicates with a database management system that supports the linking of external files, (see col. 20, lines 28-32) and column 7, lines 15-17;

"a native object store for storing said object externally to said database" as to keep large data objects stored as files in a file system and link these references to these external files from the database, (see col. 2, lines 14-16), "said object capable of being edited independently of said related metadata in a loose transaction model" as the file management system includes a check in

function which receives the updated file, and saves the updated file saves the updated file under a new name different from the original, generates new metadata for the updated file which transactionally updated file with its metadata, (see col. 4, lines 53-59), and (col. 3, lines 43-53);

"a database mediator for obtaining said ~~handle using said reference~~ encrypted access token to directly access or manipulate said external object" as a file manager when executed by the computer system that accesses files maintained in file storage and communicates with a database management system that supports the linking of external files, (see col. 20, lines 28-32) and column 7, lines 15-17;

"means for obtaining a version number embedded in said ~~handle~~ hash value of said encrypted access token" as the reference file A(1) is referred to as a versioned file, in which file A(1) is backed up in total, the backup operation of the new version file A(2) would involve backing up only the modified portions with respect to A(1), (see col. 5, lines 63-67). Burns does not explicitly disclose steps means for comparing said embedded version number with a version number of a latest committed version of said externally stored object to determine if said ~~handle~~ encrypted access token refers to a current version of said externally stored object. Guturu discloses step of comparing a version number of the data record to a version number of the data update, (see col. 2, lines 39-41), and column 4, lines 5-11-37; and also column 6, lines 47-51. It would have been obvious to one ordinary skill in the art to modify the combined teachings of Burns with Guturu with steps comparing said embedded version number with a version number of a latest committed version of said externally stored object to determine if said ~~handle~~ encrypted access token refers to a current version of said externally stored object. Such modification would allow the teachings of Burns and Guturu to improve the accuracy and the

reliability of the method of maintaining data consistency in a loose transaction model, and to provide database synchronicity without the use of synchronizing messages and thus avoiding high on the links between the databases, and to process the data update requests and maintain the same data content at the two databases, (see Guturu, col. 3, lines 4-6; and col. 3, lines 9-10).

[Remainder of page intentionally left blank]

Conclusion

6. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

CONTACT INFORMATION

7. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jean B Fleurantin whose telephone number is 703-308-6718. The examiner can normally be reached on 7:30-6:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John B Breene can be reached on 703-305-9790. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Jean Bolte Fleurantin

May 20, 2004

SHAHID ALAM
PRIMARY EXAMINER